

CLAIMS:

1           1. A system for generating floating-point  
2 test-cases for verifying the operation of a  
3 floating-point arithmetic unit, the system comprising a  
4 processing unit which includes:

5           (a) an exponent generator, for generating  
6 floating-point exponents;

7           (b) a significand generator, for generating  
8 floating-point significands; and

9           (c) a fixed-point generator coupled to said  
10 exponent generator and to said significand  
11 generator;

12           wherein said processing unit is configured to  
13 receive a specified arithmetic operation, a specified  
14 rounding mode, at least one input operand mask, and an  
15 output result mask; and wherein said processing unit is  
16 configured to output a set of floating-point numbers  
17 which includes at least one input operand compatible with  
18 said at least one input operand mask, and an output  
19 result compatible with said output result mask; and  
20 wherein said output result corresponds to said specified  
21 arithmetic operation on said at least one input operand  
22 for said specified rounding mode.

1           2. A program of instructions in data storage  
2 executable by a machine for emulating the system of claim  
3 1.

1           3. A system for generating floating-point  
2 test-cases for verifying the operation of a

3 floating-point arithmetic unit, the system comprising a  
4 processing unit which includes:

5 (a) an exponent generator, for generating  
6 floating-point exponents;

7 (b) a significand generator, for generating  
8 floating-point significands; and

9 (c) a fixed-point generator coupled to said  
10 exponent generator and to said significand  
11 generator;

12 wherein said processing unit is configured to  
13 receive a specified arithmetic operation selected from a  
14 group that includes addition and subtraction, a specified  
15 rounding mode, a first input operand mask, a second input  
16 operand mask, and an output result mask; and wherein said  
17 processing unit is configured to output a set of  
18 floating-point numbers which includes a first input  
19 operand compatible with said first input operand mask, a  
20 second input operand compatible with said second input  
21 operand mask, and an output result compatible with said  
22 output result mask; and wherein said output result  
23 corresponds to said specified arithmetic operation on  
24 said first input operand and said second input operand  
25 for said specified rounding mode.

1 4. A program of instructions in data storage  
2 executable by a machine for emulating the system of claim  
3 3.

1 5. The system of claim 3, wherein said fixed-point  
2 generator has two addends and a carry sequence  
3 representing the carries from the addition of successive

10078441-022002

4 digits of said addends, wherein said carry sequence is  
5 compatible with a carry sequence mask.

1 6. The system of claim 3, said significand  
2 generator further comprising:

3 (d) an addition significand generator, for  
4 generating floating-point significands for said  
5 addition operation; and

6 (e) a subtraction significand generator, for  
7 generating floating-point significands for said  
8 subtraction operation.

1 7. The system of claim 3, wherein said first input  
2 operand has a first input operand exponent, said second  
3 input operand has a second input operand exponent, and  
4 said output result has an output result exponent, said  
5 exponent generator further comprising:

6 (d) a definite exponent generator, for generating  
7 floating-point exponents wherein said output  
8 result exponent is a definite amount different  
9 from either of said first input operand  
10 exponent and said second input operand  
11 exponent; and

12 (e) an indefinite exponent generator, for  
13 generating floating-point exponents wherein  
14 said output result exponent is not a definite  
15 amount different from either of said first  
16 input operand exponent and said second input  
17 operand exponent.

1           8. The system of claim 3, wherein said exponent  
2 generator is a biased exponent generator, for generating  
3 biased floating-point exponents.

1           9. The system of claim 8, wherein said first input  
2 operand has a first input operand biased exponent, said  
3 second input operand has a second input operand biased  
4 exponent, and said output result has an output result  
5 biased exponent, said biased exponent generator further  
6 comprising:

7           (d) a definite biased exponent generator, for  
8 generating biased floating-point exponents  
9 wherein said output result biased exponent is a  
10 definite amount different from either of said  
11 first input operand biased exponent and said  
12 second input operand biased exponent; and

13          (e) an indefinite biased exponent generator, for  
14 generating biased floating-point exponents  
15 wherein said output result biased exponent is  
16 not a definite amount different from either of  
17 said first input operand biased exponent and  
18 said second input operand biased exponent.

1           10. The system of claim 8, further comprising an  
2 unbiased exponent shift calculator for computing an  
3 unbiased exponent shift from a biased exponent shift.

1           11. A method of seeking a solution, if a solution  
2 exists, to a specified mathematical condition, wherein  
3 the solution is used in constructing a floating-point  
4 test-case for verifying the operation of a floating-point

5 arithmetic unit, wherein a complete generated test case  
6 is a set of floating-point numbers for a specified  
7 arithmetic operation and a specified rounding mode, and  
8 wherein a generated test case includes at least one input  
9 operand and an output result; and wherein an input  
10 operand is compatible with an operand mask, and the  
11 output result is compatible with an output result mask;  
12 the method comprising the steps of:

- 13 (a) preparing a list of choices upon which the  
14 solution is based;  
15 (b) testing whether said list of choices is empty;  
16 (c) outputting, if said list of choices is empty,  
17 that no solution exists;  
18 (d) randomly choosing, if said list of choices is  
19 not empty, a choice of said list as a  
20 selection;  
21 (e) searching for a solution to the specified  
22 mathematical condition, based on said  
23 selection;  
24 (f) outputting, if said searching was successful,  
25 said solution;  
26 (g) erasing, if said searching was not successful,  
27 said selection from said list; and  
28 (h) repeating step (a) through step (g) until  
29 outputting occurs.

1 12. A program of instructions in data storage  
2 executable by a machine for performing the method of  
3 claim 11.

1 13. A method of seeking a solution, if a solution  
2 exists, to a specified mathematical condition, wherein  
3 the solution is used in constructing a floating-point  
4 test-case for verifying the operation of a floating-point  
5 arithmetic unit, wherein a complete generated test case  
6 is a set of floating-point numbers for a specified  
7 arithmetic operation selected from a group including  
8 addition and subtraction, and for a specified rounding  
9 mode, and wherein a generated test case includes a first  
10 input operand, a second input operand, and an output  
11 result; and wherein the first input operand is compatible  
12 with a first input operand mask, the second input operand  
13 is compatible with a second input operand mask, and the  
14 output result is compatible with an output result mask;  
15 the method comprising the steps of:

- 16 (a) preparing a list of choices upon which the  
17 solution is based;  
18 (b) testing whether said list of choices is empty;  
19 (c) outputting, if said list of choices is empty,  
20 that no solution exists;  
21 (d) randomly choosing, if said list of choices is  
22 not empty, a choice of said list as a  
23 selection;  
24 (e) searching for a solution to the specified  
25 mathematical condition, based on said  
26 selection;  
27 (f) outputting, if said searching was successful,  
28 said solution;  
29 (g) erasing, if said searching was not successful,  
30 said selection from said list; and

31 (h) repeating step (a) through step (g) until  
32 outputting occurs.

1 14. A program of instructions in data storage  
2 executable by a machine for performing the method of  
3 claim 13.

1 15. The method of claim 13, wherein said list of  
2 choices contains an exponent shift.

1 16. The method of claim 13, wherein the solution is  
2 a set of floating-point numbers.

1 17. The method of claim 13, wherein the solution is  
2 an exponent.

1 18. The method of claim 13, wherein the solution is  
2 a significand.

1 19. The method of claim 18, wherein said list of  
2 choices contains a tails triplet.

1 20. A method of generating a set of fixed-point  
2 numbers containing a first addend, a second addend, and a  
3 sum, wherein the first addend is compatible with a first  
4 addend mask, the second addend is compatible with a  
5 second addend mask, the sum is compatible with a sum  
6 mask, and wherein the addition of the first addend and  
7 the second addend results in a carry sequence of carry  
8 bits, wherein each carry bit has a unique index in the  
9 carry sequence, wherein the carry sequence is compatible  
10 with a carry sequence mask and wherein each carry bit has  
11 a value in the group consisting of 0, 1, and 2, and

12 wherein there exists a boundary index in the carry  
13 sequence corresponding to the lowest index of a carry bit  
14 having the value 2; the method comprising the steps of:

- 15 (a) constructing a list of possible boundary  
16 indices;
- 17 (b) testing whether said list is empty;
- 18 (c) outputting, if said list is empty, that no  
19 solution exists;
- 20 (d) randomly choosing, if said list is not empty, a  
21 boundary index from said list as a selection;
- 22 (e) searching for a carry sequence based on said  
23 selection, which is compatible with the carry  
24 sequence mask;
- 25 (f) erasing, if said searching was not successful,  
26 said selection from said list;
- 27 (g) constructing, if said searching was successful,  
28 a first addend compatible with the first addend  
29 mask, a second addend compatible with the  
30 second addend mask, and a sum compatible with  
31 the sum mask;
- 32 (h) outputting said first addend, said second  
33 addend, said sum, and said carry sequence; and
- 34 (i) repeating step (a) through step (h) until  
35 outputting occurs.

1 21. A program of instructions in data storage  
2 executable by a machine for performing the method of  
3 claim 20.